REMARKS

Claims 58-77 were examined. All claims were rejected. No claims have been amended, added or cancelled in this application.

Applicant wishes to thank the Examiner for courtesy of the telephone interview on January 12, 2003, in which Applicants and the Examiner discussed the current claims and the prior art presented in the response.

The Examiner rejected claims 58-77 under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,694,354 to Elg in view of U.S. Patent No. 6,704,824 to Goodman. However, Goodman only qualifies as prior art under 35 U.S.C. § 102(e) because it issued after Applicants' effective filing date. Applicants do not admit that Goodman is prior art and reserve the right to swear behind the reference at a later date.

The Elg reference is drawn to a system for connecting peripheral devices to various hosts (Elg, Abstract). The device includes a partial pointer (e.g. URL) that identifies the type of general driver to be transferred (Elg, Fig. 2, Col. 3, lines 31-36). From this partial pointer, the host can produce a complete pointer with the host type information (Elg, Fig. 3, Col. 3, lines 37-50). Thus, each of Elg's devices provides its own device type information. The host can then use this pointer to download a device driver for the peripheral device, from an external site or peripheral. (Elg, Col. 2, lines 49-51).

The partial pointer provided by the peripheral device in Elg is missing the operating system/platform identification, which is added by the host device. (Elg, Col. 4, line 12-14). Elg then uses the completed pointer to download the device driver from a web site, FTP site, or peripheral. (Elg, Col. 4, lines 15-19). However, the Examiner notes that Elg does not teach or suggest probing the host device. For this, the Examiner relies on Goodman.

Goodman discloses a communications adapter emulating itself as an another type of device to an attached computer based on the operating system of

the attached computer (Goodman, Col. 4, lines 15-24). For example, the device emulates a keyboard or a compact disk read only memory (CD-ROM) device for the Windows 98 operating system (Goodman, Col. 4, lines 25-28; Col. 5, lines 35-40). The device sends a first identification to the attached computer to identify itself as the emulated device (Goodman, Col. 2, lines 26-30; Col. 4, lines 19-21). The emulated device then sends commands to the attached computer to upload a driver (Goodman, Col. 4, lines 38-47). After the driver is installed, the device sends the attached computer a second identification identifying the device as the communications adapter (Goodman, Col. 2, lines 33-36; Col. 4, line 66 – Col. 5, line 11). The above examples are given for a Windows 98 operating system (Goodman, Col. 5, lines 57-58).

Goodman further discloses that the communication device can include software that is "adapted to different and possibly multiple operating systems" (Goodman, Col. 5, lines 58-61). The communications adapter detects the attached computer's operating system during "the initial [universal serial bus (USB)] communication exchange" (Goodman, Col. 5, lines 61-63). However, Goodman does not disclose how the communications adapter detects the attached computer's operating system using the USB protocol.

Claim 58 recites:

> A method of interaction between a client device and a host device to be performed when the client device is connected to the host device, the method comprising:
>
> establishing a bidirectional communication channel between the client device and the host device using a handshake command/response;
>
> negotiating a reliable stream protocol connection between the client device and the host device, data for the reliable stream protocol connection to flow over the bidirectional communication channel;
>
> probing the host device by the client device with a probe message to identify the type of host device;
>
> identifying the host device type by the client device with the handshake response, the handshake response transmitted by the host device in response to receiving the probe message;
>
> transmitting executable information selected according to an identity of the host device from the client device to the host device over the reliable stream protocol connection and receiving a file handle for the executable information at the host device;
>
> invoking execution by the client of the executable information at the host device using the file handle; and
>
> entering a listening mode to receive a message sent by the executable information executing at the host device.

(Claim 58, emphasis added). Claim 58 recites "probing the host device by the client device with a probe message to identify the type of host device." Furthermore, claim 58 recites "identifying the host device type by the client device with the handshake response, the handshake response transmitted by the host device in response to receiving the probe message." The Examiner correctly states that Elg does not disclose these claim elements and relies on Goodman as disclosing them.

The Examiner asserts that because Goodman's communications adapter knows the computer's operating system, that Goodman implicitly discloses that the "host computer informs the [communications adapter] of its traits" (Office Action, p. 3). Since Goodman merely discloses that the host computer's operating system identification occurs during the initial USB exchange, Goodman does not specifically discuss what type of messages are used by the host computer to inform the communications adapter of the host computer's traits.

Because Goodman is silent of the type of messaging system used for the host computer's traits, Goodman cannot teach or suggest that the communication adapter probes the host computer for the host's computer's type. Thus, Goodman does not teach or disclose "probing the host device by the client device ... to identify the host device" as claimed in claim 58.

The Examiner asserts that since the communications adapter "knows which commands to use to control the [communications adapter], it is implicit that the host computer is identified to the [communications adapter]." However, even if the host computer is identified to Goodman's communications adapter, Goodman does not disclose how the host computer is identified to Goodman's communications adapter. Thus, Goodman does not teach or suggest "identifying the host device type by the client device with the handshake response, the handshake response transmitted by the host device in response to receiving the probe message" as claimed in claim 58.

Furthermore, Goodman does not provide details on how the USB initial exchange is used to detect operating system information. Goodman discloses that the communications adapter detects the host computer's operating system during the initial USB exchange. However, as is known in the art, the host computer does not pass the operating system information to USB-attached devices during the initial USB setup. See, e.g., "USB in a Nutshell – Chapter 6 – USB Requests", p. 1-2, where a SETUP packet is used for configuration of a USB device (submitted in an Information Disclosure Sheet herewith). The USB SETUP packet does not include operating system type information. Furthermore, during the process of attaching a USB device to a host, the host sends requests to the USB device, with the USB device responding to each request of the host by returning information and taking other requested actions (See, "USB Complete: Enumeration, p.1, submitted in an Information Disclosure Sheet herewith). Thus, as is known in the art, within the USB protocol the host controls the process of attaching a USB device and the host does not send the operating system type information to the USB device. Goodman does not teach or suggest

an alternate use of the USB initial exchange in which such information is exchanged. Therefore, Goodman does not teach or suggest to one of skill in the art the use of USB initial exchange protocol to detect operating system information.

Applicants' claim 58 recites probing the host device and handshake response transmission, which is supported in the Specification, for example, at Figure 4A; page 36, line 18 – page 37, line 37; and page 41, lines 28-30. Therefore, claims 58 and claims 59-65 that depend on claim 58 are not rendered obvious by Goodman and Elg.

Claim 66 recites:

> An apparatus comprising:
> a physical interface manager to detect when the apparatus is connected to a host, to probe the host in order to identify the type of host;
> a protocol manager to negotiate a reliable bidirectional data communication channel to the host;
> a driver uploader to identify the type of the host based on a handshake response received from the host in response to the host receiving the probe, transmit a driver appropriate for the host type to the host over the reliable bidirectional data communication channel, receive a file handle for the driver at the host, and invoke the driver at the host using the file handle; and
> a command server to respond to commands from the driver.

(Claim 66, emphasis added). Claim 66 recites a physical interface manager "to probe the host in order to identify the type of host" and a device uploaded "to identify the type of the host based on a handshake response received from the host in response to the host receiving the probe." As discussed above, neither Goodman nor Elg teach or suggest these claim elements. Therefore, claims 66 and claims 67-74 that depend on claim 66 are not rendered obvious by Goodman and Elg.

Claim 75 recites:

A client device designed to be coupled to a host device, the client device comprising:

a physical interface manager to detect when the client device is connected to the host device;

a protocol manager to negotiate a reliable bidirectional data communication channel to the host device;

a driver uploader to identify the type of the host device based on a handshake response received from the host in response to the host receiving a probe, based on data received during the negotiation of the data communication channel, transmit a driver appropriate for the host type to the host device over the reliable bidirectional data communication channel.

(Claim 75, emphasis added). Claim 75 recites a device uploaded "to identify the type of the host based on a handshake response received from the host in response to the host receiving a probe." As discussed above, neither Goodman nor Elg teach or suggest these claim elements. Therefore, claims 75 and claims 76-77 that depend on claim 75 are not rendered obvious by Goodman and Elg.

CONCLUSION

In view of the foregoing, it is believed that claims 58-77, patentably define the subject invention over the prior art of record, and are in condition for allowance and such action is earnestly solicited at the earliest possible date.

If the Examiner believes that a telephone conference would be useful in moving the application forward to allowance, the Examiner is encouraged to contact the undersigned at (408)720-8300.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Dated: February 17, 2009

Eric S. Replogle, Reg. No. 52,161